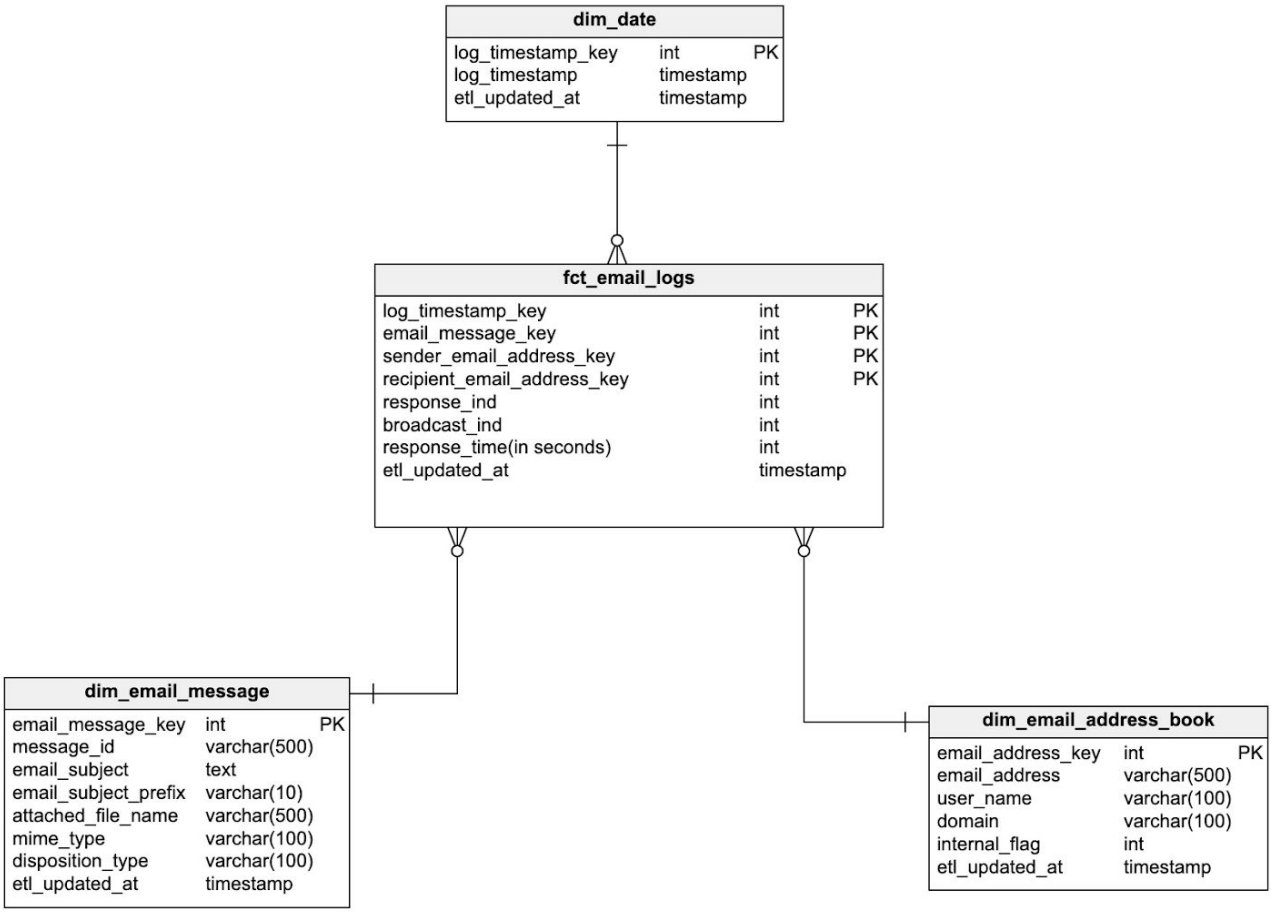
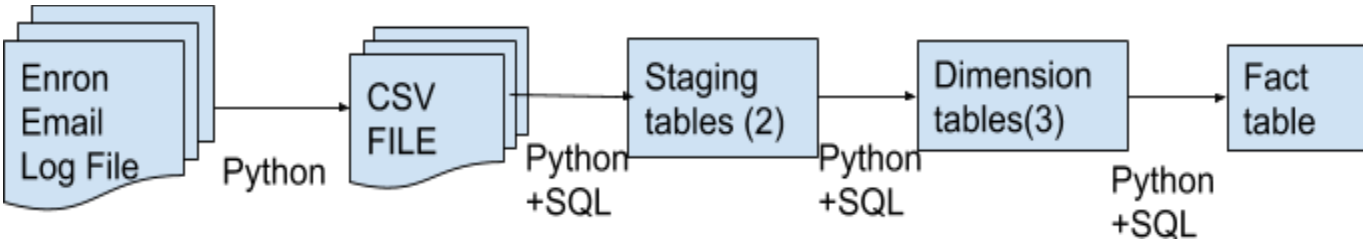
Enron Email Log - Data Modeling & Analysis

Data Model:



Data Flow:



Design Considerations:

1. Why did I choose star schema for this project?
    - ❖ This project is focused to perform BI analysis and hence dimensional model would fit better for serving all future needs that may arise.There is no space limitation and hence we are good to choose star schema design.
    - ❖ It's intuitive for data analysts to understand the dimensional model.
    - ❖ Advanced SQL skills are not required to analyse this data model.Most of the analysis can be performed by joining facts with dimensions and applying filters. So this model would serve as a good fit for organizations having decentralized IT/BI teams [Teams with basic SQL skills can use this model]
    - ❖ The trade off here is that "the data engineer writes complex sql queries to load the data so that it will be easy for analysts to consume".
    - ❖ This model can serve for most of the analytical queries on top of the enron email logs (not just for coding challenge questions)

2. Granularity of fact table:
    - ❖ Enron_db.**fct_email_logs** captures the log records at the most granular level (one record per email message per sender per recipient)
    - ❖ As the data is captured at the most granular level, It's easy for us to roll the data up to required summary level depending on the future requirements.

3. Measures of fact table:

- Non additive measure response_time is added to fact table.
- This measure will be populated only for the email messages that were responses to original email messages. Otherwise this measure will simply be NULL
- **Assumption made in deriving this response_time measure:**
  - **I found some cases during testing where there were more than one original email message for a given response email.In those cases, i took the most recent original email for calculating response_time.**
  - **I also enforced a condition that the timestamp of the response email should be greater than or equal to the timestamp of the original email.**

4. Why did I design two staging tables?

- Staging tables are used to load the dimension and fact tables from log files on a daily basis.
- In general, Comma Separated Values in a column is not a good fit to serve BI purpose requirements.It's difficult to process CSV columns in a relational table using just SQL.
- That's the reason the second fact table is designed to capture the pivoted recipient email addresses.

5. Debugging Friendly:

- "Etl_updated_at" is added to all the tables, so that it will be easy for debugging.

6. Scalability:

- In real time, this model can be deployed to handle huge volume of production data sets by scheduling the python file load.py
- Fact table is designed to have only data type integers and Primary Key enabled. This will help us in running efficient queries against it.
- enron_db.dim_email_address_book can be used as a conformed dimension across different use cases.It helps to analyse on the email address , users and domain and internal/external email movement.

- For time-being, enron_db.dim_date is designed to capture only the logged timestamp dates.In real time, dim_date will be designed to capture all the calendar dates and can be used as a conformed dimension.

7. Error handling:

- Currently all the I/O and SQL errors are handled smoothly in python.
- Constraints are enabled to check if we are loading the same file again and again.
- In case of errors , it will be displaying the error message along with function name in the console.
- In future, We can implement an error file class with which we can write all the logs to error file

Future Improvements

- Implementing Error File Class to write errors to a file
- Implementing dim_date as a rolling calendar date dimension
- Implementing matlab plot in python to visualize the insights.